Development of a Related Document Retrieval System and Evaluation of the system using NTCIR-1

Hiroshi Umemoto[†], Tsutomu Kuramochi[†], Yasuhiro Ishitobi[†], Masakazu Tateno[†]

[†]Fuji Xerox Co., Ltd.

{umemoto,kuramoch,ishitobi,tateno}@rsl.crl.fujixerox.co.jp

1. Introduction

Related document retrieval is a function to output related documents in the order of the similarities to the documents that are relevant to a user's need. We have developed a related document retrieval system.

Our retrieval system handles structured documents with a TRIE-formed word index file. The index includes index words that are tagged by document field information.

In general, a document retrieval system is evaluated in its recall rate, precision and so on. We have done a experiment in order to evaluate our retrieval system using NTCIR-1. Actually, next two retrieval cases have been considered:

(1) A user constructs retrieval expressions manually

(2) The system constructs retrieval expressions automatically

In this paper, we describe the method of processing structured documents, the algorithm of the related document retrieval and the evaluation of the system using NTCIR-1.

2. Processing structured documents

2.1 Overview

In general indexing methods, index words are stored in multiple tables, and each table includes index words that appear in the same document field layer. In searching, a retriever has to search index words from all tables.

While, in our retrieval system, an index word is tagged by document field information, and is stored in a TRIE-formed word index file.

The overview of the processing structured documents is represented in Figure 1. In this paper, a structured document is not only described in SGML-like format, but also embedded implicit structure tags which are represented by specified strings. And a CSV(Comma Separated Value) document is also a structured document. In a CSV document, a row is a subject of the retrieval, and a column is a document field.

2.2 Encoding of the document field information

Document field information is encoded in a binary byte array. Document structure tags are extracted from the structured documents, then each tag is assigned to specified binary byte array. Length of an assigned byte array is not fixed, but is according to the depth of its document field layer. Figure 2 is an example of encoding of document field information.

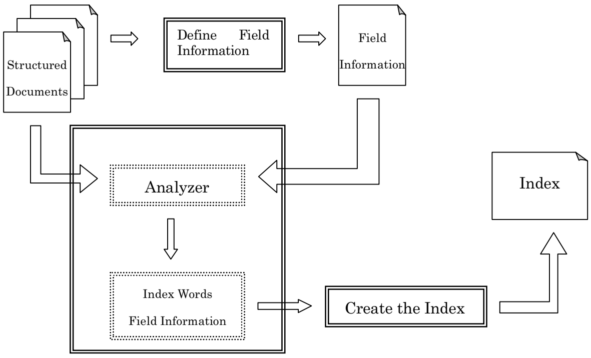
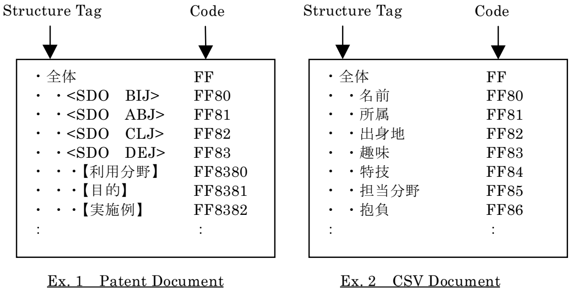


Figure 1. Overview of the processing structured documents

An index word is encoded in the combination of its word string, a delimiter byte and a byte array of its document field information. For example, an index word "テスト" which appears in the document field "【目的】" is encoded in a binary byte array "A5 C6 A5 B9 A5 C8 FF 83 81"(in the hexadecimal format). The character code is EUC-kanji, and the delimiter byte is "FF".



Structure Tag	Code
・全体	FF
・　・<SDO　BIJ>	FF80
・　・<SDO　ABJ>	FF81
・　・<SDO　CLJ>	FF82
・　・<SDO　DEJ>	FF83
・　・・【利用分野】	FF8380
・　・・【目的】	FF8381
・　・・【実施例】	FF8382
:	:

Ex. 1　Patent Document

Structure Tag	Code
・全体	FF
・　・名前	FF80
・　・所属	FF81
・　・出身地	FF82
・　・趣味	FF83
・　・特技	FF84
・　・担当分野	FF85
・　・抱負	FF86
:	:

Ex. 2　CSV Document

Figure 2.　Examples of Document Field Information

2.3 Extraction of index words

Subjective documents written in Japanese are applied to a Japanese morphological analysis. We have developed a Japanese morphological analyzer that is based on two-level morphology, and apply the analyzer to the retrieval system.

From the output of the analyzer, content words are registered in the index. And compound words are also registered. Compound words consist of at least two noun words that place in the neighborhood. For example, A, B and C are three noun word and "ABC" are described in a document, then in the index, six words A, B, C, AB, BC and ABC are registered.

2.4 Method of retrieval for structured documents

An example of the TRIE-formed index is described in Figure 3.

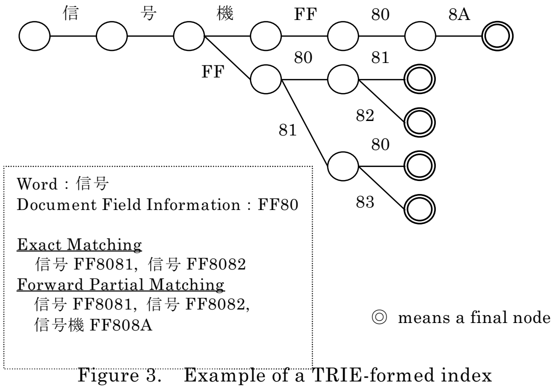


Figure 3. Example of a TRIE-formed index

(1) Exact matching

A user specifies a search word and a document field where the word appears.

Then the retrieval system begins to search index words which consist of the search word and the document field information from the TRIE-formed index file. In searching, the system search es not only exactly matched index words, but also forward partially matched ones that are found in the lower document field layers.

(2) Forward partial matching

A user specifies a sub-string of a search word and a document field where the word appears.

Then first, the retrieval system begins to search index words from the TRIE-formed index file that include the sub-string. Second, the system selects index words that include the specified document field information.

3. Algorithm of the related document retrieval

3.1 Overview

Document retrieval systems have been proposed which are based on TF-IDF method and word vector space method.

While, the algorithm of our related document retrieval is based on the concept of mutual information in the information theory.

3.2 Calculation of the similarity of a word

First, whole subjective documents are encoding as a information source, whether a index word exists in a document or not. Then in the information source, information entropy H1 is calculated that an index word A exists in any document. And in case of a document set Q is given as a retrieval condition, conditioned information entropy H2 is calculated. Information I of A which is calculated when the document set Q is given is the difference of information entropy H1 and H2. In other words, information I of an index word A is mutual information of H1 and H2.

$$H1 = - \log_2 \mathrm{prob}(A)$$

$$H2 = - \log_2 \mathrm{prob}(A|Q)$$

$$\begin{aligned} MI &= H1 - H2 \\ &= \log_2 (\mathrm{prob}(A|Q) / \mathrm{prob}(A)) \\ &= \log_2 (\mathrm{prob}(A, Q) / (\mathrm{prob}(A) \cdot \mathrm{prob}(Q))) \\ &= \log_2 (\beta \cdot D / (\alpha \cdot DQ)) \end{aligned}$$

In above, prob(A) is a probability that the word A exists in a document, prob(Q) is a ratio of the count of documents in Q against that of the information source, prob(A|Q) is a probability that A exists in Q, prob(A,Q) is a probability that a document includes A and also belongs to Q. D is the total count of documents in the information source, α is the count of documents where word A appears, β is the count of documents where word A appears and that belong to Q.

All index words in Q are considered as related words and the information of a related word is considered as a related score, then the related score of each related word is calculated. And if it is assumed that the probability that each related word exists in a document is independent, then the information of the case that multiple related words exist in the same document is the sum of information of related words.

3.3 Calculation of the similarity of a document

Now documents including at least one related word are considered as a related document, and the related score of a related document is defined as the sum of related scores of related words in the document.

In the implementation of the actual retrieval system, a related score of A in case that Q is given is calculated as the following equation:

$$MI'(A, Q) = \beta^2 / \alpha$$

Related Words	β^2	α	Score
装置	9^2	9,000	0.009
車両	8^2	200	0.32
自動車	4^2	200	0.08
ナビゲーション	2^2	100	0.04
:	:	:	:

And words are weighted according to the document field where words appear. For example, a word appearing in the title field is weighted two times as one appearing in the author field.

Tag		Weight
REC		
ACCN		
TITL TYPE=″kanji″		2
TITE TYPE=″alpha″		2

AUPK TYPE=″kanji″		
AUPE TYPE=″alpha″		
CONF TYPE=″kanji″		
CNFE TYPE=″alpha″		
CNFD		
ABST TYPE=″kanji″		1
ABSE TYPE=″alpha″		1
ABST.P		
ABSE.P		
KYWD TYPE=″kanji″		5
KYWE TYPE=″alpha″		5
SOCN TYPE=″kanji″		1
SOCE TYPE=″alpha″		1

4. Evaluation of the system using NTCIR-1

4.1 Method of the experiment

In an experiment, next four retrieval methods are considered:

(Method 1) An user constructs an retrieval expression manually, then executes a full text searching

(Method 2) A related document retrieval is executed with the result of Method 1, and relevance feed-backs are executed repeatedly

(Method 3) A related document retrieval is executed automatically with a retrieval demand

(Method 4) A related document retrieval is executed with the result of Method 3

The target theme of this experiment is number 0029.

Actual methods are followed:

(Method 1) A full text search was executed with the following retrieval expression, then 56 results were obtained.

(位置計測＋位置測定＋位置検出) * 物体

(Method 2) Each result of Method 1 was checked, and 12 documents were picked up as relevant. And, a related document retrieval was executed with the 12 relevant documents, then there were 15 relevant documents in the result of the related document retrieval, so the retrieval process was terminated.

(Method 3) A related document retrieval is executed automatically with a retrieval demand

(Method 4) A related document retrieval is executed with the result of Method 3

4.2 Result

Correct results were judged by the correct set A. Table 1 is the relation between the count of extracted relevant documents and the average precision, and Figure 4 is the graph of the relation between the recall rate and the interpolated precision.

	Extracted	Relevant	Average Precision
Method 1	56	22	0.0080
Method 2	1000	62	0.1300
Method 3	1000	72	0.1710
Method 4	1000	69	0.1480

Table 1. relation between the count of extracted relevant documents

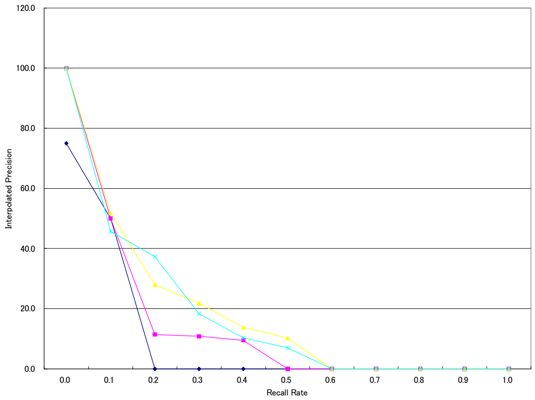


Figure 4. relation between the recall rate and the interpolated precision

5. Conclusion

From Table 1, Method 3 is the best in the count of extracted documents and the average precision.

6. References

[1] 増市,山浦,小山,舘野,｢形態素解析を用いた全文検索システムとその応用｣, 情報処理学会自然言語処理, 102-3, pp.17-24(1994.7)

[2] G. Salton et al. : ”Introduction to Modern Information Retrieval”, McGraw-Hill, New York, 1983.

[3] 丹羽:｢動的な共起解析を用いた対話的文書検索支援｣, 情報処理学会研究報告 96-NL-115.

Columns on Last Page Should Be Made Equal Length